

[illegible]

```

LL               IIIII
LL               IIIII
LL               III
LL               III
LL               III
LL               III
LL               III
LL               III
LL               III
LL               III
LL               III
LL               III
LL               III
LL               III
LLLLLLLLLLLL    IIIII
LLLLLLLLLLLL    IIIII

                SSSSSSS
                SSSSSSS
                SS
                SS
                SS
                SS
                SSSSSS
                SSSSSS
                SS
                SS
                SS
                SS
                SSSSSSS
                SSSSSSS

```

(1)	487	Macros for Loadable Services
(1)	707	INHEXCP - Inhibited CHMK or CHME code handling
(1)	817	MP\$ASTEXIT - AST EXIT SYSTEM SERVICE FOR SECONDARY PROCESSOR
(1)	872	CHANGE MODE DETECTED ERROR HANDLING
(1)	922	Filtered Change Mode to Kernel Dispatcher
(1)	991	CHANGE MODE TO KERNEL DISPATCHER
(1)	1734	REGION 2 OF SYS. SERV. VECTOR DEFINITIONS

```
0000 1 :
0000 2 : Version: 'V04-000'
0000 3 :
0000 4 :
0000 5 :
00000001 5 : .MCALL MFPR
0000 6 : MPSWITCH = 1
0000 7 : .NLIST CND
0000 17 : .TITLE MPCMOD - MULTIPROCESSING KERNEL SYS SRV DISPATCHER FOR SECONDARY
0000 19 : .IDENT 'V04-000'
0000 20 :
0000 21 :
0000 22 : *****
0000 23 : *
0000 24 : * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
0000 25 : * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
0000 26 : * ALL RIGHTS RESERVED. *
0000 27 : *
0000 28 : * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0000 29 : * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
0000 30 : * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
0000 31 : * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0000 32 : * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
0000 33 : * TRANSFERRED. *
0000 34 : *
0000 35 : * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
0000 36 : * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
0000 37 : * CORPORATION. *
0000 38 : *
0000 39 : * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
0000 40 : * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
0000 41 : *
0000 42 : *
0000 43 : *****
0000 44 :
0000 45 : D. N. CUTLER 22-JUN-76
0000 46 :
0000 47 : MODIFIED BY:
0000 48 :
0000 49 : V03-041 LJK0287 Lawrence J. Kenah 27-Jun-1984
0000 50 : Add R5 to entry mask for $SCANEXH system service.
0000 51 :
0000 52 : V03-040 LMP0239 L. Mark Pilant, 23-Apr-1984 9:21
0000 53 : Change $CHKPRO from an exec mode service to a kernel mode
0000 54 : service. This was made necessary by the $CHKPRO (internal
0000 55 : entry point) interface change.
0000 56 :
0000 57 : V03-039 MMD0250 Meg Dumont, 27-Feb-1984 17:49
0000 58 : Add support for $MTACCESS installation specific accessibility
0000 59 : routine
0000 60 :
0000 61 : V03-038 DAS0001 David Solomon 20-Feb-1984
0000 62 : Implement new design for RMS echo SYS$INPUT to SYS$OUTPUT
0000 63 : (vs V03-019). Echo is now performed by a caller's mode AST
0000 64 : routine declared in RMS\RMSEXAMS. Change INCB/DECB of FAB/RAB
0000 65 : busy bit to BISB/BICB, now that we have room.
0000 66 :
0000 67 : V03-037 SSA0004 Stan Amway 28-Dec-1983
```



0000	68	:	For \$SETPFM, changed number of parameters from 1 to 4
0000	69	:	and changed entry mask to save R2-R11.
0000	70	:	
0000	71	:	V03-036 TMK0002 Todd M. Katz 19-Nov-1983
0000	72	:	The entry point for \$ASCTOID can no longer be reached as a
0000	73	:	branch destination from the executive mode dispatcher.
0000	74	:	A temporary entry point (EXE\$ASCTOID) has been placed within
0000	75	:	this module, and a JMP is made from it to the real system
0000	76	:	service entry point (EXE\$ASCTOID).
0000	77	:	
0000	78	:	Also, change the entry mask for SYS\$TRNLOG, so that R8 is
0000	79	:	now saved.
0000	80	:	
0000	81	:	V03-035 TMK0001 Todd M. Katz 22-Oct-1983
0000	82	:	The entry points for \$FINISH_RDB and \$IDTOASC can no
0000	83	:	longer be reached as branch destinations from the executive
0000	84	:	mode dispatcher. Temporary entry points (EXE\$FINISH_RDB and
0000	85	:	EXE\$IDTOASC) have been placed within this module, and from
0000	86	:	each a JMP is made to the real system service entry points
0000	87	:	(EXE\$FINISH_RDB and EXE\$IDTOASC).
0000	88	:	
0000	89	:	V03-034 PRB0254 Paul Beck 15-Sep-1983 14:49
0000	90	:	(1) Correct the way synchronous CJF services are defined.
0000	91	:	(2) Define loadable RUF services.
0000	92	:	
0000	93	:	V03-033 WMC0029 Wayne Cardoza 31-Aug-1983
0000	94	:	Loadable services should not be unconditionally inhibited.
0000	95	:	Add an alternate CHMx argument to LDBSRV.
0000	96	:	
0000	97	:	V03-032 DWT0125 David W. Thiel 22-Aug-1983
0000	98	:	Remove CHECKARGLIST and calls to same.
0000	99	:	
0000	100	:	V03-031 MKL0167 Mary Kay Lyons 19-Aug-1983
0000	101	:	Generate loadable service vector for CJF\$GETCJI.
0000	102	:	
0000	103	:	V03-030 KBT0578 Keith B. Thompson 8-Aug-1983
0000	104	:	Add parameter to \$FILESCAN
0000	105	:	
0000	106	:	V03-029 RAS0178 Ron Schaefer 29-Jul-1983
0000	107	:	Add code to detect the AST/non-AST RMS FAB/RAB race
0000	108	:	condition where an RMS operation is initiated while
0000	109	:	the user FAB/RAB is still waiting for completion of
0000	110	:	previous operation.
0000	111	:	
0000	112	:	V03-028 WMC0028 Wayne Cardoza 29-Jun-1983
0000	113	:	Add CJF services.
0000	114	:	
0000	115	:	V03-027 WMC0027 Wayne Cardoza 23-Jun-1983
0000	116	:	Make old logical name services "all mode".
0000	117	:	Changes to image activator vectors.
0000	118	:	
0000	119	:	V03-026 JWH0222 Jeffrey W. Horn 2-May-1983
0000	120	:	Add LDBSRV macro for vector definitions of loadable
0000	121	:	services.
0000	122	:	
0000	123	:	V03-025 DMW4035 DMWalp 26-May-1983
0000	124	:	Intergate new logical name structures.

0000	125	:	
0000	126	:	
0000	127	:	V03-024 LMP0109 L. Mark Pilant, 28-Apr-1983 15:53
0000	128	:	Make \$CHKPRO an EXEC mode system service to allow examination
0000	129	:	of various system data structures.
0000	130	:	
0000	131	:	V03-024 RAS0147 Ron Schaefer 28-APR-1983
0000	132	:	Add \$FILESCAN. Add R8 and R9 to \$SETPRN register mask.
0000	133	:	
0000	134	:	V03-023 JLV0244 Jake VanNoy 27-APR-1983
0000	135	:	Add \$BRKTHRU. Change \$BRDCST to all mode service.
0000	136	:	\$BRDCST now uses \$BRKTHRU to do real work.
0000	137	:	
0000	138	:	V03-022 LMP0099 L. Mark Pilant, 13-Apr-1983 19:15
0000	139	:	Add the \$CHKPRO system service.
0000	140	:	
0000	141	:	V03-021 ACG0319 Andrew C. Goldstein, 21-Mar-1983 13:51
0000	142	:	Add \$GRANTID and \$REVOKID services
0000	143	:	
0000	144	:	V03-020 JLV0234 Jake VanNoy 1-MAR-1983
0000	145	:	Add \$BRKTHRU service.
0000	146	:	
0000	147	:	V03-019 RAS0120 Ron Schaefer 25-Feb-1983
0000	148	:	Add support to echo SYS\$INPUT to SYS\$OUTPUT.
0000	149	:	This involves examining the return code from RMS for \$GET;
0000	150	:	if the special status RMSS ECHO (not returned to users)
0000	151	:	is found, then create a RAB on the caller's stack and
0000	152	:	execute a \$PUT operation to echo the line.
0000	153	:	A certain amount of RMS synchronization code was
0000	154	:	shuffled around in order to make room for this.
0000	155	:	
0000	156	:	V03-018 ACG0317 Andrew C. Goldstein, 22-Feb-1983 15:16
0000	157	:	Fix off-by-one in kernel arg vector
0000	158	:	
0000	159	:	V03-017 RSH0004 R. Scott Hanna 10-Feb-1983
0000	160	:	Added \$ASCTOID, \$FINISH_RDB, and \$IDTOASC to system service list
0000	161	:	
0000	162	:	V03-016 RNG0016 Rod N. Gamache 1-Feb-1983
0000	163	:	Added \$GETLKI to system service list
0000	164	:	
0000	165	:	V03-015 WMC0015 Wayne Cardoza 12-Jan-1983
0000	166	:	Put back accidentally deleted space holder for RMS synchronization.
0000	167	:	
0000	168	:	V03-014 DMW4023 DMWalp 7-Jan-1983
0000	169	:	Added \$CRELNT, \$CRELNM, \$DELLNM and \$TRNLNM
0000	170	:	
0000	171	:	V03-013 KDM0033 Kathleen D. Morse 13-Dec-1982
0000	172	:	Correct usage of an interlocked instruction to flush
0000	173	:	the hardware cache queue.
0000	174	:	
0000	175	:	V03-012 ROW0146 Ralph O. Weber 6-DEC-1982
0000	176	:	Insert routine header comments for INHEXCP, CHECKARGLIST,
0000	177	:	and EXE\$CMODKRNLY (MPSS\$CMODKRNLY). Move things around so
0000	178	:	that EXE\$CMODKRNLY (MPSS\$CMODKRNLY) header comments are near
0000	179	:	EXE\$CMODKRNLY (MPSS\$CMODKRNLY) and ASTEXIT comments are near
0000	180	:	ASTEXIT. Make basic kernel-mode .PSECT definition for Y\$CMODK
0000	181	:	or MP\$CMOD1 immediately after executive mode code so that new
			code can be inserted in a way that preserves routine headers.



```
0000 182 : conditional assembly, and .PSECT definitions. Backout ROW145,
0000 183 : and in its place, correct conditional assembly of BGEQU 10$
0000 184 : after ACCVIO RET so that it is assembled only for MPCMOD and
0000 185 : so that it is located before ACCVIO RET. Change PCB address
0000 186 : lookup at KERDSP in MPCMOD to use CTL$GL_PCB so that it works
0000 187 : correctly regardless of which processor executes it.
0000 188 :
0000 189 : V03-011 ROW0145 Ralph O. Weber 29-NOV-1982
0000 190 : Move EXE$EXCPTN (and MPS$EXCPTN) to before ASTEXIT (or
0000 191 : MPSS$ASTEXIT) in an attempt to make branch destinations in
0000 192 : EXE$CMODKRNL reach.
0000 193 :
0000 194 : V03-010 KDM0030 Kathleen D. Morse 18-Nov-1982
0000 195 : Add logic to MPCMOD that allows the primary to execute
0000 196 : secondary-specific code, without turning into a secondary.
0000 197 :
0000 198 : V03-009 MLJ0099 Martin L. Jack, 20-Oct-1982 19:42
0000 199 : Complete V03-002 by correcting mode and argument count of
0000 200 : $SNDJBC and removing temporary stubs.
0000 201 :
0000 202 : V03-008 RIH0001 Richard I. Hustvedt 1-Jun-1982
0000 203 : Correct handling of AST queue by secondary processor to
0000 204 : avoid losing some AST notifications by incorrectly computing
0000 205 : PHD$B_ASTLVL.
0000 206 :
0000 207 : V03-007 KDM0018 Kathleen D. Morse 30-Sep-1982
0000 208 : Add MPSWITCH logic to create a kernel system service
0000 209 : dispatcher for the secondary processor of an 11/782.
0000 210 :
0000 211 : V03-006 STJ3028 Steven T. Jeffreys 26-Sep-1982
0000 212 : Added $ERAPAT system service vector.
0000 213 :
0000 214 : V03-005 DWT0058 David Thiel 11-Aug-1982
0000 215 : Eliminate use of R2 while waiting for service
0000 216 : completion.
0000 217 :
0000 218 : V03-004 JWH0001 Jeffrey W. Horn 26-Jul-1982
0000 219 : Add new RMS service, RMSRUHNDLR, an un-documented service
0000 220 : which acts as the Recovery Unit handler for RMS.
0000 221 :
0000 222 : V03-003 PHL0102 Peter H. Lipman 16-Jul-1982
0000 223 : Fix new SYNCH logic to always return SS$_NORMAL,
0000 224 : not access IOSB if error from service, and return
0000 225 : error status from $SETEF if event flag cluster went away
0000 226 :
0000 227 : V03-002 PHL0101 Peter H. Lipman 17-Jun-1982
0000 228 : Add $SYNCH system service and fix $QIOW and $ENQW to use the
0000 229 : new code for waiting for the combination of EFN and IOSB
0000 230 :
0000 231 : Improve readability of conditionals.
0000 232 :
0000 233 : Add $GETDVIW, $GETJPIW, $GETSYIW, $SNDJBC, $SNDJBCW, and
0000 234 : $UPDSECW. All the waiting versions use common code.
0000 235 :
0000 236 :
0000 237 :
0000 238 : CHANGE MODE SYSTEM SERVICE DISPATCHER
```

```
0000 239 :  
0000 240 : MACRO LIBRARY CALLS  
0000 241 :  
0000 242 :  
0000 243 $ACBDEF ;DEFINE AST CONTROL BLOCK OFFSETS  
0000 244 $CHFDEF ;DEFINE CONDITION HANDLING OFFSETS  
0000 245 $ENQDEF ;DEFINE ENQ SYSTEM SERVICE ARGS  
0000 246 $GETDVIDEF ;DEFINE GETDVI SYSTEM SERVICE ARGS  
0000 247 $GETJPIDEF ;DEFINE GETJPI SYSTEM SERVICE ARGS  
0000 248 $GETLKIDEF ;DEFINE GETLKI SYSTEM SERVICE ARGS  
0000 249 $GETSYIDEF ;DEFINE GETSYI SYSTEM SERVICE ARGS  
0000 250 $IPLDEF ;DEFINE INTERRUPT PRIORITY LEVELS  
0000 252 $LCKDEF ;DEFINE INTERLOCK BITS  
0000 254 $PCBDEF ;DEFINE PCB OFFSETS  
0000 255 $PHDDEF ;DEFINE PHD OFFSETS  
0000 256 $PRDEF ;DEFINE PROCESSOR REGISTERS  
0000 257 $PSLDEF ;DEFINE PROCESSOR STATUS FIELDS  
0000 258 $RABDEF ;DEFINE RMS RAB FIELDS  
0000 259 $RPBDEF ;DEFINE REBOOT PARAMETER BLOCK  
0000 260 $QIODEF ;DEFINE QIO SYSTEM SERVICE ARGS  
0000 261 $SGNDEF ;DEFINE SYSGEN PARAMETERS  
0000 262 $SNDJBCDEF ;DEFINE SNDJBC SYSTEM SERVICE ARGS  
0000 263 $SSDEF ;DEFINE SYSTEM STATUS VALUES  
0000 264 $SYNCHDEF ;DEFINE SYNCH SYSTEM SERVICE ARGS  
0000 265 $UPDSECDDEF ;DEFINE UPDATE SECTION SYS SRV ARGS  
0000 266 :  
0000 267 : LOCAL EQUATES  
0000 268 :  
00000001 0000 269 CAT0 = 100  
00000080 0000 270 CAT7 = 107  
00000081 0000 271 DEF_MASK = CAT0!CAT7 ;INHIBIT FOR 'ALL' AND 'NOT EXIT'  
00000080 0000 272 EXC_MASK = CAT7 ;INHIBIT ONLY FOR 'ALL' CASE  
0000 273 :  
0000 274 : LOCAL MACROS  
0000 275 :  
0000 276 GSYSSRV - GENERATE SYSTEM SERVICE ENTRY VECTOR  
0000 277 :  
0000 278 GSYSSRV SRVNAME,MODE,NARG,REGISTERS,MASK,NOSYNC  
0000 279 :  
0000 280 WHERE:  
0000 281 SRVNAME - SERVICE NAME LESS ANY PREFIX (SYS$,EXE$,RMS$)  
0000 282 MODE - MODE DESIGNATOR FOR SERVICE (K,E,ALL,R)  
0000 283 NARG - REQUIRED NUMBER OF ARGUMENTS  
0000 284 REGISTERS - REGISTER SAVE LIST  
0000 285 MASK - SERVICE INHIBIT MASK(BIT SET IN CAT INHIBITS)  
0000 286 NOSYNC - NON-ZERO IF RMS SYNCHRONIZATION CODE NOT TO BE INCLUDED  
0000 287 :  
0000 288 :  
0000 289 .MACRO GSYSSRV,SRVNAME,MODE,NARG,REGS,MASK=DEF_MASK,NOSYNC  
0000 290 .IF NDF,RMSSWITCH  
0000 291 .IF DF,LIBSWITCH  
0000 292 .PSECT $$$0000,QUAD  
0000 293 .IFF  
0000 294 .PSECT $$$000,QUAD  
0000 295 .ENDC  
0000 296 .ALIGN QUAD  
0000 297 .IF DF LIBSWITCH
```



```

0000 298 SYSS' SRVNAME::
0000 299 .IFF
0000 300 .IF NDF,MPSWITCH
0000 301 .WORD ^M<REGS>
0000 302 SRVNAME' MASK = ^M<REGS>
0000 303 .IFTF ;MPSWITCH
0000 304 .IF B NOSYNC
0000 305 SRV'MODE SRVNAME,NARG,MASK
0000 306 .IFF
0000 307 SRV'MODE SRVNAME,NARG,MASK,NOSYNC
0000 308 .ENDC
0000 309 .ENDC ;MPSWITCH
0000 310 .IFT
0000 311 .BLKL 2
0000 312 .ENDC
0000 313 .IFF
0000 314 SRV'MODE SRVNAME,NARG,MASK
0000 315 .ENDC
0000 316 .ENDM GSYSSRV
0000 317
0000 318 ::
0000 319 GCOMPSRVB - GENERATE COMPOSITE SYSTEM SERVICE ENTRY VECTOR BEGIN
0000 320 ::
0000 321 GCOMPSRVB SRVNAME,REGISTER_MASK[,PREFIX]
0000 322 ::
0000 323 WHERE:
0000 324 SRVNAME - SERVICE NAME LESS ANY PREFIX (SYSS, EXES)
0000 325 REGISTER_MASK - SYMBOLIC REGISTER MASK, E.G QIO MASK
0000 326 PREFIX - IF SUPPLIED, THE PREFIX FOR THE SERVICE NAME.
0000 327 IF OMITTED, "SYSS" IS ASSUMED.
0000 328 ::
0000 329
0000 330 .MACRO GCOMPSRVB,SRVNAME,REGMSK,PREFIX=SYSS
0000 331 .IF NDF,MPSWITCH
0000 332 .IF NDF,RMSSWITCH
0000 333 .IF DF,LIBSWITCH
0000 334 .PSECT $$$0000,QUAD
0000 335 .IFF
0000 336 .PSECT $$$000,QUAD
0000 337 .ENDC
0000 338 .ALIGN QUAD
0000 339 .IF DF LIBSWITCH
0000 340 .IIF NOT_BLANK, <SRVNAME>,-
0000 341 'PREFIX' SRVNAME::
0000 342 .IFF
0000 343 .ENABL LSB
0000 344 COMPSTR=
0000 345 .IIF NOT_BLANK, <REGMSK>,-
0000 346 .WORD <REGMSK>
0000 347 .ENDC
0000 348 .ENDC
0000 349 .ENDC ;MPSWITCH
0000 350 .ENDM GCOMPSRVB
0000 351
0000 352 ::
0000 353 GCOMPSRVE - GENERATE COMPOSITE SYSTEM SERVICE ENTRY VECTOR END
0000 354 ::

```

```
0000 355 :      GCOMPSRVE      QUADWORDS
0000 356 :
0000 357 :      WHERE:
0000 358 :      QUADWORDS - NUMBER OF QUADWORDS TO RESERVE FOR VECTOR
0000 359 :
0000 360 :
0000 361 :      .MACRO GCOMPSRVE,QUADS
0000 362 :      .IF NDF,MPSWITCH
0000 363 :      .IF NDF,RMSSWITCH
0000 364 :      .IF DF,LIBSWITCH
0000 365 :      .BLKQ QUADS
0000 366 :      .IFF
0000 367 COMPsize=-COMPSTRT
0000 368 :      .IF GE,QUADS*8-COMPsize
0000 369 :      .BLKB QUADS*8-COMPsize
0000 370 :      .IFF
0000 371 :      .ERROR ; VECTOR EXCEEDS ALLOCATED SIZE ;
0000 372 :      .ENDC
0000 373 :      .DSABL LSB
0000 374 :      .ENDC
0000 375 :      .ENDC
0000 376 :      .ENDC ;MPSWITCH
0000 377 :      .ENDM GCOMPSRVE
0000 378 :
0000 379 :
0000 380 :
0000 381 :      SRVK - GENERATE ENTRY FOR KERNEL MODE SERVICE
0000 382 :
0000 383 :      SRVK SRVNAME,NARG,MASK
0000 384 :
0000 385 :
0000 386 :      .MACRO SRVK,SRVNAME,NARG,MASK
0000 387 :      .IF NDF,RMSSWITCH
0000 388 :      .IF DF,MPSWITCH
0000 389 CMK$C_'SRVNAME==KCASCTR
0000 390 :      .IFF ;MPSWITCH DEFINED
0000 391 CMK$C_'SRVNAME=KCASCTR
0000 392 :      CHMK #SRVNAME
0000 393 :      RET
0000 394 :      .PSECT Y$MODKN,BYTE
0000 395 :      .=KCASCTR
0000 396 :      ASSUME NARG LE 127
0000 397 :      .BYTE NARG
0000 398 :      .PSECT Y$MODKX,BYTE
0000 399 :      .=KCASCTR
0000 400 :      .BYTE MASK
0000 401 :      .PSECT Y$MODK,BYTE
0000 402 :      .SIGNED_WORD EXES'SRVNAME-KCASE+2
0000 403 :      .IFTF ;MPSWITCH
0000 404 SRVNAME=KCASCTR
0000 405 KCASCTR=KCASCTR+1
0000 406 :      .ENDC ;MPSWITCH
0000 407 :      .ENDC
0000 408 :      .ENDM SRVK
0000 409 :
0000 410 :
0000 411 :      SRVE - GENERATE ENTRY FOR EXECUTIVE MODE SERVICE
```

```
0000 412 :
0000 413 :
0000 414 .MACRO SRVE,SRVNAME,NARG,MASK
0000 415 .IF NDF,MPSWITCH
0000 416 .IF NDF,RMSSWITCH
0000 417 CMESC_'SRVNAME=ECASCTR
0000 418 CHME #SRVNAME
0000 419 RET
0000 420 .PSECT Y$MODEN,BYTE
0000 421 .=ECASCTR
0000 422 ASSUME NARG LE 127
0000 423 .BYTE NARG
0000 424 .PSECT Y$MODEX,BYTE
0000 425 .=ECASCTR
0000 426 .BYTE MASK
0000 427 .PSECT Y$MODE,BYTE
0000 428 .SIGNED_WORD EXES'SRVNAME-ECASE+2
0000 429 .ENDC
0000 430 SRVNAME=ECASCTR
0000 431 ECASCTR=ECASCTR+1
0000 432 .ENDC ;MPSWITCH
0000 433 .ENDM SRVE
0000 434 :
0000 435 :
0000 436 :
0000 437 :
0000 438 .MACRO RMSSRV SRVNAME NARG=1,REGS=<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>,-
0000 439 MASK,NOSYNC=0
0000 440 GSYSSRV SRVNAME,R,NARG,<REGS>,MASK,NOSYNC
0000 441 .ENDM RMSSRV
0000 442 :
0000 443 :
0000 444 :
0000 445 .MACRO SRVR SRVNAME,NARG,MASK,NOSYNC
0000 446 .IF NDF,MPSWITCH
0000 447 .IF NDF,RMSSWITCH
0000 448 CMESC_'SRVNAME=RCASCTR
0000 449 CHME #SRVNAME
0000 450 .IF EQ NOSYNC
0000 451 .IIF GT <.+2-RMSSYNC>-127,-
0000 452 RMSSYNC=RMSWBR ;RESET BRANCH DESTINATION
0000 453 RMSWBR=.
0000 454 BRB RMSSYNC
0000 455 .IFF
0000 456 RET
0000 457 .ENDC
0000 458 .PSECT Y$MODEN,BYTE
0000 459 .=RCASCTR
0000 460 ASSUME NARG LE 127
0000 461 .BYTE NARG
0000 462 .PSECT Y$MODEX,BYTE
0000 463 .=RCASCTR
0000 464 .BYTE MASK
0000 465 .IFF
0000 466 .PSECT $$$RMSVEC,BYTE,NOWRT
0000 467 .SIGNED_WORD RMSS'SRVNAME-RCASE+2
0000 468 .ENDC
```



```
0000 469 SRVNAME=RCASCTR
0000 470 RCASCTR=RCASCTR+1
0000 471 .ENDC :MPSWITCH
0000 472 .ENDM SRVR
0000 473
0000 474 :
0000 475 : SRVALL - GENERATE ENTRY FOR ALL MODE SERVICE
0000 476 :
0000 477
0000 478 .MACRO SRVALL,SRVNAME,NARG,MASK
0000 479 .IF NDF,MPSWITCH
0000 480 .IF NDF,RMSSWITCH
0000 481 JMP @EXES,SRVNAME+2
0000 482 .ENDC
0000 483 .ENDC :MPSWITCH
0000 484 .ENDM SRVALL
0000 485
```

```
0000 487      .%BTTL  Macros for Loadable Services
0000 488
0000 489      :
0000 490      LDBSRV - Generate Loadable Service Vector
0000 491      :
0000 492      LDBSRV PREFIX,SRVNAME,MODE,REGS,SYN_EFN,SYN_IOSB,ALT_CHMX
0000 493      :
0000 494      Where:
0000 495          PREFIX      - Prefix for system service vector entry point name
0000 496          SRVNAME      - Service name less any prefix (SYSS,CJFS, etc.)
0000 497          MODE        - Mode designator for service (K,E,ALL)
0000 498          REGS        - Register save list
0000 499          SYN_EFN      - Event flag argument number for $SYNCH
0000 500          SYN_IOSB    - IOSB argument number for $SYNCH
0000 501          ALT_CHMX     - Use same CHMX number as this service
0000 502      :
0000 503
0000 504      .MACRO LDBSRV,PREFIX,SRVNAME,MODE,REGS,SYN_EFN,SYN_IOSB,ALT_CHMX
0000 505      .IF NDF,RMSSWITCH
0000 506      .IF NDF,MPSWITCH
0000 507      .IF DF,LIBSWITCH
0000 508      .PSECT $$$0000,QUAD
0000 509      .ALIGN QUAD
0000 510      PREFIX''SRVNAME::
0000 511      .IF BLANK SYN_EFN
0000 512      .BLKL 2
0000 513      .IFF
0000 514      .BLKL 4
0000 515      .ENDC
0000 516      .IFF
0000 517      .PSECT $$$000,QUAD
0000 518      .ALIGN QUAD
0000 519      .WORD *M<REGS>
0000 520      SRVNAME' MASK = *M<REGS>
0000 521      LVEC_'MODE PREFIX,SRVNAME,SYN_EFN,SYN_IOSB,ALT_CHMX
0000 522      .ENDC
0000 523      .ENDC : MPSWITCH
0000 524      .ENDC : RMSSWITCH
0000 525      .ENDM LDBSRV
0000 526
0000 527      :
0000 528      LVEC_K - Kernel Mode Loadable System Service Vector
0000 529      :
0000 530      LVEC_K PREFIX,SERVICE,EFN,IOSB
0000 531      :
0000 532
0000 533      .MACRO LVEC_K,PREFIX,SERVICE,EFN,IOSB,ALT_CHMX
0000 534      .IF BLANK ALT_CHMX
0000 535      CMKSC_'SERVICE = PREFIX'KCASCTR
0000 536      .IFF
0000 537      CMKSC_'SERVICE = ALT_CHMX
0000 538      .ENDC
0000 539      CMK #SERVICE
0000 540      .IF NOT BLANK EFN
0000 541      PUSRL #EFN
0000 542      PUSHL #IOSB
0000 543      JMP @EXESLDB_SYNCH
```

```
0000 544 .IFF
0000 545 RET
0000 546 .ENDC
0000 547 .IF BLANK ALT_CHMK
0000 548 SERVICE = PREFIX'KASCTR
0000 549 PREFIX'KASCTR = PREFIX'KASCTR + 1
0000 550 .IFF
0000 551 SERVICE = ALT_CHMK
0000 552 .ENDC
0000 553 .ENDM LVEC_K
0000 554
0000 555 ..
0000 556 LVEC_E - Exec Mode Loadable System Service Vector
0000 557 ..
0000 558 LVEC_E PREFIX,SERVICE,EFN,IOSB
0000 559 ..
0000 560
0000 561 .MACRO LVEC_E,PREFIX,SERVICE,EFN,IOSB,ALT_CHME
0000 562 .IF BLANK ALT_CHME
0000 563 CMESC_'SERVICE = PREFIX'ECASCTR
0000 564 .IFF
0000 565 CMESC_'SERVICE = ALT_CHME
0000 566 .ENDC
0000 567 CHME #SERVICE
0000 568 .IF NOT BLANK EFN
0000 569 PUSHL #EFN
0000 570 PUSHL #IOSB
0000 571 JMP @#EXESLDB_SYNCH
0000 572 .IFF
0000 573 RET
0000 574 .ENDC
0000 575 RET
0000 576 .IF BLANK ALT_CHME
0000 577 SERVICE = PREFIX'ECASCTR
0000 578 PREFIX'ECASCTR = PREFIX'ECASCTR + 1
0000 579 .IFF
0000 580 SERVICE = ALT_CHME
0000 581 .ENDC
0000 582 .ENDM LVEC_E
0000 583
0000 584 ..
0000 585 LVEC_ALL - Mode of caller Loadable System Service Vector
0000 586 ..
0000 587 LVEC_ALL PREFIX,SERVICE,EFN,IOSB
0000 588 ..
0000 589 .MACRO LVEC_ALL,PREFIX,SERVICE,EFN,IOSB,ALT_CHMK
0000 590 JMP @#EXES'SERVICE
0000 591 .IF NOT BLANK EFN
0000 592 ERROR ; SYNCH NOT ALLOWED FOR ALL-MODE SERVICES
0000 593 .ENDC
0000 594 .ENDM LVEC_ALL
0000 595
0000 596
0000 694
0000 695
0000 696
0000 697
```



MPCMOD  
V04-000

J 15  
- MULTIPROCESSING KERNEL SYS SRV DISPATCH 16-SEP-1984 02:08:16 VAX/VMS Macro V04-00  
Macros for Loadable Services 5-SEP-1984 03:40:37 [SYS.SRC]CMODSSDSP.MAR;1

Page 12  
(1)

```
      0000 698 :  
      0000 699 : Establish .PSECT for kernel-mode servicing code which follows  
      0000 700 :  
00000000 704 .PSECT MP$CMOD1,QUAD
```

MP  
VO

```
0000 707 .SBTTL INHEXCP - Inhibited CHMK or CHME code handling
0000 708
0000 709 :+
0000 710 :
0000 711 INHEXCP - Inhibited CHMK or CHME code handling
0000 712
0000 713 FUNCTIONAL DESCRIPTION:
0000 714
0000 715 When the ability to use specified system services is inhibited
0000 716 via the SSETSSF system service, this routine receives control
0000 717 when an attempt to execute an inhibited system service occurs.
0000 718
0000 745 The exception condition is returned to the primary processor for exception
0000 746 handling.
0000 747
0000 748 INPUTS:
0000 749
0000 750 R1 = SS error code (SS$_INHCHMK or SS$_INHCHME)
0000 751 00(SP) = Change mode parameter code
0000 752 04(SP) = Saved PC of exception
0000 753 08(SP) = Saved PSL of exception
0000 754
0000 755 ENVIRONMENT:
0000 756
0000 757 This code executes on the secondary processor.
0000 758 If interrupted at any point, may continue on the primary processor.
0000 759
0000 760 :-
0000 767 INHEXCP:
51 DD 0000 768 PUSHL R1 ;PUSH THE EXCEPTION CODE
04 DD 0002 769 PUSHL #4 ;PUSH THE NUMBER OF ARGUMENTS
0004 773 IFPRIMARY <JMP G^EXES$REFLECT> ;IF PRIMARY, THEN CONTINUE RIGHT ALONG
001D 774 ;IF SECONDARY, RETURN PROCESS TO PRIMARY
7E 10 AE 02 18 EF 001D 775 EXTZV #PSL$V_CURMOD,#PSL$S_CURMOD,16(SP),-(SP) ;CREATE PSL WITH PREV
6E 6E 16 9C 0023 776 ROTL #PSL$V-PRVMOD,(SP),(SP) ;MODE CORRECT AND CURRENT MODE = KERNEL
00000000'GF 9F 0027 777 PUSHAB G^EXES$REFLECT ;REFLECT THE EXCEPTION
FFD0' 31 002D 778 BRW MPSS$MPSCHED2 ;AND RETURN PROCESS TO PRIMARY
```

```
0030 817 .SBTTL MPSS$ASTEXIT - AST EXIT SYSTEM SERVICE FOR SECONDARY PROCESSOR
0030 818
0030 819 :+
0030 820 : FUNCTIONAL DESCRIPTION:
0030 821 : This is the AST exit system service routine for the secondary processor
0030 822 : only. It clears the AST active bit for the appropriate mode, in the
0030 823 : process' PCB and then sets a new AST level (both in the PHD and the
0030 824 : secondary's processor register). Because an AST may be delivered by
0030 825 : the primary while the secondary is executing this code, the routine
0030 826 : is repeated until the head of the AST queue is stable.
0030 827
0030 828
0030 829 : INPUTS:
0030 830 :
0030 831 : (SP) - PC at time of interrupt
0030 832 : 4(SP) - PSL at time of interrupt
0030 833
0030 834 : ENVIRONMENT:
0030 835 :
0030 836 : Executes on the secondary processor.
0030 837 : If interrupted at any point, may continue on the primary processor.
0030 838
0030 839 :-
0030 840
00000000 841 .PSECT MP$CMOD2,BYTE
50 04 AE 02 18 EF 0000 842 MPSS$ASTEXIT:
54 53 DD 0006 843 EXTZV #PSL$V_CURMOD,#PSL$S_CURMOD,4(SP),R0 ; Get previous mode
52 DD 0008 844 PUSHL R4 ; Save register
54 0000'CF D0 000A 845 PUSHL R3 ; Save register (This is faster)
00 0C A4 50 E7 0011 846 PUSHL R2 ; Save register (than a PUSHR.)
50 10 A4 DE 0014 847 MOVL W*MPSS$GL_CURPCB,R4 ; Get address of current process' PCB
52 04 D0 001D 848 SETIPL #IPL$ SYNCH ; Disable system events
51 60 D0 0020 849 BBCCI R0,PCB$B_ASTACT(R4),10$ ; Clear AST active bit for this mode
51 50 D1 0023 850 10$: MOVAL PCB$B_ASTQFL(R4),R0 ; Get address of AST queue
52 13 D4 0026 851 MOVL #4,R2 ; Assume null AST level
52 0D 0028 852 MOVL (R0),R1 ; Get flink
52 06 19 002A 853 CMPL R0,R1 ; Is the queue empty?
52 08 8F 88 002D 854 BEQL 20$ ; Br on yes, set null AST level
52 0A 8F 88 002F 855 CLRL R2 ; Assume kernel mode
52 0B A1 FC 8F 88 002A 856 ASSUME ACB$V_KAST EQ 7
52 0B A1 FC 8F 88 002D 857 TSTB ACB$B_RMOD(R1) ; Check for kernel AST
52 0B A1 FC 8F 88 002F 858 BLSS 20$ ; Br if not kernel AST
52 0B A1 FC 8F 88 002F 859 BICB3 #^C<3>,ACB$B_RMOD(R1),R2 ; Get request mode
52 0B A1 FC 8F 88 0035 860 20$: MOVL PCB$B_PHD(R4),R3 ; Get address of PHD
52 0B A1 FC 8F 88 0039 861 MTPR R2,#PR$ ASTLVL ; Set ASTLVL register
52 0B A1 FC 8F 88 003C 862 MOV B R2,PHD$B_ASTLVL(R3) ; Set ASTLVL in PHD
52 0B A1 FC 8F 88 0041 863 BBSSI #LCK$V_INTERLOCK,W*MPSS$GL_INTERLOCK,30$ ; Flush cache queue
52 0B A1 FC 8F 88 0047 864 30$: CMPL (R0),RT ; Has the head of the queue changed?
52 0B A1 FC 8F 88 004A 865 BNEQ 10$ ; Yes, repeat ASTLVL computation
52 0B A1 FC 8F 88 004C 866 MOVQ (SP)+,R2 ; Restore registers
52 0B A1 FC 8F 88 004F 867 POPL R4 ; Restore register
52 0B A1 FC 8F 88 0052 868 REI ; Return from interrupt
00000030 869 .PSECT MP$CMOD1,QUAD
```



```
0030 872 .SBTTL CHANGE MODE DETECTED ERROR HANDLING
0030 873 :+
0030 874 : ACCVIO - ACCESS VIOLATION DETECTED IN ARGUMENT LIST
0030 875 : INSARG - INSUFFICIENT ARGUMENTS SUPPLIED FOR SERVICE
0030 876 : SSFAIL - ABNORMAL STATUS RETURNED BY SERVICE ROUTINE
0030 877 :
0030 878 : THESE ROUTINES TAKE THE APPROPRIATE ACTION TO RETURN THE ERROR INDICATION
0030 879 : TO THE ORIGINAL CALLER.
0030 880 :
0030 881 :-
0030 882 .ENABL LSB
0030 883 ACCVIO:
0030 884 MOVL SP,FP ;SET FRAME POINTER BEFORE RET
0033 885 CMPW RO,#KASCTR ;IS THIS AN UNRECOGNIZED CODE?
0038 887 BGEQU 10$ ;YES, NOT NECESSARILY ACCVIO
003A 892 MOVZWL #SS$_ACCVIO,RO ;SET ACCESS VIOLATION
003D 893 RET
003E 894
003E 895 KINSARG: CMPW RO,#KASCTR ;IS THIS AN UNRECOGNIZED CODE?
0043 896 10$: BGEQU KERDSP ;YES, NOT NECESSARILY INSARG
0045 900 MOVZWL #SS$_INSFARG,RO ;SET INSUFFICIENT NUMBER OF ARGUMENTS
004A 902 RET
004B 903 SRVEXIT:
004B 904 BLBC RO,SSFAIL ;SERVICE EXIT
004E 905 SRVREI: REI ;BR IF ABNORMAL COMPLETION
004F 907 MPSS$EXCPN::
004F 911 .WORD 0 ;SYSTEM SERVICE EXCEPTION
0051 915 SECBUG_CHECK SSRVEXCEPT,FATAL ;ENTRY MASK
0056 917 SSFAIL: BITL #7,RO ;UNEXPECTED SYSTEM SERVICE EXCEPTION
0059 918 BEQL SRVREI ;TEST SEVERITY FIELD
005B 919 BRW SSFAILMAIN ;IF EQL WARNING
005E 920 .DSABL LSB ;GOTO MAIN SSFAIL LOGIC
```

0057'8F 50 SE D0 0030 884  
50 09 1E 0033 885  
50 0C 3C 0038 887  
003A 892  
003D 893  
003E 894  
0057'8F 50 B1 003E 895  
72 1E 0043 896  
50 0114 8F 3C 0045 900  
004A 902  
004B 903  
08 50 E9 004B 904  
02 004E 905  
004F 907  
0000 004F 911  
50 07 D3 0051 915  
F3 13 0056 917  
FFF5' 31 0059 918  
005B 919  
005E 920

```
005E 922 .SBTTL Filtered Change Mode to Kernel Dispatcher
005E 923
005E 924
005E 928 MPSS$CMODKRN LX - Secondary Filtered Change Mode to Kernel Dispatcher
005E 930
005E 931 When inhibiting of user mode system service calls has been enabled via the
005E 935 SSINHIBIT SYSGEN parameter, this routine -- not MPSS$CMODKRN LX -- is called
005E 937 whenever a CHMK instruction is executed. The state of the stack on entry
005E 938 is:
005E 939
005E 940 INPUTS:
005E 941
005E 942 00(SP) = CHANGE MODE PARAMETER CODE.
005E 943 04(SP) = SAVED PC OF EXCEPTION.
005E 944 08(SP) = SAVED PSL OF EXCEPTION.
005E 945
005E 946 00(AP) = NUMBER OF SYSTEM SERVICE CALL ARGUMENTS.
005E 947 04(AP) = FIRST ARGUMENT.
005E 948
005E 949
005E 950
005E 951 4*N(AP) = N'TH ARGUMENT.
005E 952
005E 953 OUTPUTS:
005E 954
005E 955 THE APPROPRIATE KERNEL MODE SYSTEM SERVICE IS INVOKED.
005E 956
005E 957
0000005E 964 .PSECT MP$CMOD1,QUAD
005E 966
005E 967 .ALIGN QUAD
0060 971 MPSS$CMODKRN LX::
0060 973 BICL3 8(SP),#PSL$M CURMOD,R0 ;CHECK THE PREVIOUS MODE
0069 977 BNEQ W^MPSS$CMODKRN LX ;NO CHECK NEEDED FOR NON-USER MODE
006B 979 MOVZBL (SP),R0 ;PICK UP THE CHMK CODE
006E 984 BITB G^SY$GB KMASK[R0],G^CTL$GB SSFILTER ;'AND' WITH INHIBIT MASK
007A 985 BEQL W^MPSS$CMODKRN LX ;THIS CODE IS ALLOWED
007C 987 MOVZWL #SS$ INHCHMK,R1 ;SET THE EXCEPTION CODE
0081 988 BRW INH$CP ;AND REFLECT IT
0084 989
```

50 03000000 8F 08 AE CB 0060 973  
1D 12 0069 977  
50 6E 9A 006B 979  
00000000'GF 00000000'GF 40 93 006E 984  
OC 13 007A 985  
51 04CC 8F 3C 007C 987  
FF7C 31 0081 988  
0084 989

```
0084 991 .SBTTL CHANGE MODE TO KERNEL DISPATCHER
0084 992 :+
0084 996 : MPSS$CMODKRNL - SECONDARY CHANGE MODE TO KERNEL DISPATCHER
0084 998 :
0084 999 : THIS ROUTINE IS AUTOMATICALLY VECTORED TO WHEN A CHANGE MODE TO KERNEL
0084 1000 : INSTRUCTION IS EXECUTED. THE STATE OF THE STACK ON ENTRY IS:
0084 1001 :
0084 1002 : INPUTS:
0084 1003 :
0084 1004 : 00(SP) = CHANGE MODE PARAMETER CODE.
0084 1005 : 04(SP) = SAVED PC OF EXCEPTION.
0084 1006 : 08(SP) = SAVED PSL OF EXCEPTION.
0084 1007 :
0084 1008 : 00(AP) = NUMBER OF SYSTEM SERVICE CALL ARGUMENTS.
0084 1009 : 04(AP) = FIRST ARGUMENT.
0084 1010 :
0084 1011 :
0084 1012 :
0084 1013 : 4*N(AP) = N'TH ARGUMENT.
0084 1014 :
0084 1015 : OUTPUTS:
0084 1016 :
0084 1017 : THE APPROPRIATE KERNEL MODE SYSTEM SERVICE IS INVOKED.
0084 1018 : -
0084 1019 :
0084 1020 : .ALIGN QUAD
0088 1024 MPSS$CMODKRNL::
0088 1026 :
0088 1027 :
0088 1028 :
0088 1029 :
0088 1035 : POPL R0 : REMOVE CHANGE MODE PARAMETER FROM STACK
0088 1039 : BEQL ASTEXIT : IF ZERO, AST EXIT SYSTEM SERVICE
008D 1041 : PUSHAB B*SRVEXIT : RETURN ADDRESS
0090 1042 : MOVZBL R0,R1 : BOUND RANGE OF CHMK CODES TO 0,255
0093 1043 : : AND 256 BYTES ACCESSIBLE FROM B_KRNLNARG
0093 1044 : PUSHL FP : SAVE FP
0095 1048 : MOVZBL G*SYS$GB_KRNLNARG[R1],R1 : GET NUMBER OF REQUIRED ARGUMENTS
009D 1050 : PUSHL AP : SAVE AP
009F 1051 : MOVAL @#4[R1],FP : CALCULATE LENGTH OF ARGUMENT LIST
00A7 1052 : CLRQ -(SP) : PSW AND REGISTER SAVE MASK
00A9 1056 : IFNORD FP,(AP),ACCVIO1 : DECLARE ACCESS VIOLATION
00AF 1058 : MOVL SP,FP : SET FRAME POINTER FOR CALL FRAME
00B2 1059 : CMPB (AP),R1 : CHECK FOR REQUIRED NUMBER OF ARGS
00B5 1065 : BLSSU KINSARG1 : IF LSSU, INSUFFICIENT ARGUMENTS
00B7 1066 : KERDSP: MOVL G*CTL$GL_PCB,R4 : GET CURRENT PROCESS PCB ADDRESS
00BE 1067 : CMPW R0,#WAITFR : IS THIS THE WAITFR SYSTEM SERVICE?
00C3 1068 : BEQL MP$SWAITFR1 : BR ON YES, EXECUTE SYS SRV ON SECONDARY
00C5 1069 : CMPW R0,#WFLAND : IS THIS THE WFLAND SYSTEM SERVICE?
00CA 1070 : BEQL MP$SWFLAND1 : BR ON YES, EXECUTE SYS SRV ON SECONDARY
00CC 1071 : CMPW R0,#WFLOR : IS THIS THE WFLOR SYSTEM SERVICE?
00D1 1072 : REQL MP$SWFLOR1 : BR ON YES, EXECUTE SYS SRV ON SECONDARY
00D3 1073 : ADDL #8,SP : CLEAN OFF PSW AND REG SAVE MASK
00D6 1074 : POPL AP : RESTORE AP
00D9 1075 : POPL FP : RESTORE FP
00DC 1076 : MOVL R0,(SP) : REPLACE CHMK ON STACK OVER RET ADR
00DF 1077 : IFPRIMARY <JMP G*EXE$CMODKRNL> : IF PRIMARY, THEN CONTINUE RIGHT ALONG
```

50 8ED0 0088 1035  
7E 13 0088 1039  
BB AF 9F 008D 1041  
51 50 9A 0090 1042  
5D DD 0093 1043  
51 00000000'GF 41 9A 0095 1048  
5C DD 009D 1050  
5D 00000004 9F 41 DE 009F 1051  
7E 7C 00A7 1052  
5D 5E D0 00AF 1058  
51 6C 91 00B2 1059  
59 1F 00B5 1065  
54 00000000'GF D0 00B7 1066  
003B'8F 50 B1 00BE 1067  
4E 13 00C3 1068  
003D'8F 50 B1 00C5 1069  
4A 13 00CA 1070  
003E'8F 50 B1 00CC 1071  
46 13 00D1 1072  
5E 08 C0 00D3 1073  
5C 8ED0 00D6 1074  
5D 8ED0 00D9 1075  
6E 50 D0 00DC 1076  
00DF 1077



```
75 08 AE 02 18 EF 00F8 1078 ;IF SECONDARY, RETURN PROCESS TO PRIMARY
    6E 6E 16 9C 00F8 1079 EXTZV #PSL$V_CURMOD,#PSL$S_CURMOD,8(SP),-(SP);CREATE PSL WITH PREV
    00000000'GF 9F 00FE 1080 ROTL #PSL$V-PRVMOD,(SP),(SP);MODE CORRECT AND CURRENT MODE = KERNEL
    FEF5' 31 0102 1081 PUSHAB G*EXE$CMODKRNL;EXECUTE THE SERVICE ON PRIMARY
    0108 1082 BRW MP$MPSCHED2;AND RETURN PROCESS TO PRIMARY
    010B 1083
    010B 1084 ASTEXIT:
    F3' 11 010B 1085 BRB MP$ASTEXIT;BRANCH ASSIST
    010D 1086 ACCVIO1:
    FF20 31 010D 1087 BRW ACCVIO;BRANCH ASSIST
    0110 1088 KINSARG1:
    FF2B 31 0110 1089 BRW KINSARG;BRANCH ASSIST
    0113 1090
    0113 1091 ;
    0113 1092 ; BRANCH ASSISTS TO REACH SYSTEM SERVICES.
    0113 1093 ;
    0113 1094 MP$WAITFR1:
    FEEC' 31 0113 1095 BRW MP$WAITFR+2;BRANCH ASSIST (PAST REG SAVE MASK)
    0116 1096 MP$WFLAND1:
    FEE9' 31 0116 1097 BRW MP$WFLAND+2;BRANCH ASSIST (PAST REG SAVE MASK)
    0119 1098 MP$WFLOR1:
    FEE6' 31 0119 1099 BRW MP$WFLOR+2;BRANCH ASSIST (PAST REG SAVE MASK)
    011C 1101 KCASE:
    00000001 011C 1102 KCASCTR=1;BASE OF CHMK CASE TABLE
    011C 1209 .ALIGN QUAD;CHMK CODES START AT 1
```

```
0120 1213 :  
0120 1214 :  
0120 1215 :  
0120 1216 :  
0120 1217 :  
0120 1218 :  
0120 1219 :  
0000 1220 :  
0000 1221 :  
0000 1222 :  
0000 1223 :  
0000 1224 :  
0000 1225 :  
0000 1226 :  
0000 1227 :  
0000 1228 :  
0000 1229 :  
0000 1230 :  
0000 1231 :  
0000 1232 :  
0000 1233 :  
0000 1234 :  
0000 1235 :  
0000 1236 :  
0000 1237 :  
0000 1238 :  
0000 1239 :  
0000 1240 :  
0000 1241 :  
0000 1242 :  
0000 1243 :  
0000 1244 :  
0000 1245 :  
0000 1246 :  
0000 1247 :  
0000 1248 :  
0000 1249 :  
0000 1250 :  
0000 1251 :  
0000 1252 :  
0000 1253 :  
0000 1254 :  
0000 1255 :  
0000 1256 :  
0000 1257 :  
0000 1258 :  
0000 1259 :  
0000 1260 :  
0000 1261 :  
0000 1262 :  
0000 1263 :  
0000 1264 :  
0000 1265 :  
0000 1266 :  
0000 1267 :  
0000 1268 :  
0000 1269 :  
  
DEFINE REMAINING SERVICES  
  
GSYSSRV ADJSTK,K,3,- :ADJUST OUTER MODE STACK POINTER  
      <R2,R3,R4,R5,R6>,- :REGISTERS R2-R6  
      EXC MASK :EXCEPTION MASK  
GSYSSRV ADJWSL,K,2,- :ADJUST WORKING SET LIMIT  
      <R2,R3,R4,R5> :REGISTERS R2-R5  
GSYSSRV ALCDNP,K,4,- :ALLOCATE DIAGNOSTIC PAGE  
      <R2,R3,R4,R5,R6,R7> :REGISTERS R2-R7  
GSYSSRV ALLOC,K,4,- :ALLOCATE DEVICE  
      <R2,R3,R4,R5,R6> :REGISTERS R2-R6  
GSYSSRV ASCFC,K,4,- :ASSOCIATE COMMON EVENT FLAG CLUSTER  
      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> :REGISTERS R2-R11  
GSYSSRV ASCTIM,ALL,3,- :CONVERT TO ASCII TIME  
      <R2,R3,R4,R5,R6> :REGISTERS R2-R6  
GSYSSRV ASSIGN,K,4,- :ASSIGN I/O CHANNEL  
      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> :REGISTERS R2-R11  
GSYSSRV BINTIM,ALL,2,- :CONVERT TO BINARY TIME  
      <R2,R3,R4,R5,R6,R7,R8> :REGISTERS R2-R8  
GSYSSRV CANCEL,K,1,- :CANCEL I/O ON CHANNEL  
      <R2,R3,R4,R5,R6,R7,R8> :REGISTERS R2-R8  
GSYSSRV CANTIM,K,2,- :CANCEL TIMER REQUEST  
      <R2,R3,R4,R5> :REGISTERS R2-R5  
GSYSSRV CANWAK,K,2,- :CANCEL WAKE UP REQUESTS  
      <R2,R3,R4,R5> :REGISTERS R2-R5  
GSYSSRV CRMPSC,K,12,- :CREATE AND MAP SECTION  
      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> :REGISTERS R2-R11  
GSYSSRV CLRPAR,K,2,- :CLEAR HARD PARITY ERROR  
      <R2,R3,R4,R5> :REGISTERS R2-R5  
GSYSSRV CMEXEC,E,2,- :CHANGE MODE TO EXECUTIVE  
      <R4> :REGISTER R4  
GSYSSRV CMKRNL,K,2,- :CHANGE MODE TO KERNEL  
      <R4> :REGISTER R4  
GSYSSRV CLREF,K,1,- :CLEAR EVENT FLAG  
      <R2,R3,R4,R5> :REGISTERS R2-R5. SEE WAITFR COMMENTS.  
GSYSSRV CNTREG,K,4,- :CONTRACT REGION  
      <R2,R3,R4,R5,R6,R7> :REGISTERS R2-R7  
GSYSSRV GETPTI,K,5,- :GET PAGE TABLE INFORMATION  
      <R2,R3,R4,R5,R6,R7,R8,R9,R10> :REGISTERS R2-R10  
GSYSSRV CRELOG,ALL,4,- :CREATE LOGICAL NAME  
      <R2,R3,R4,R5,R6,R7,R8> :REGISTERS R2-R8  
GSYSSRV CREMBX,K,7,- :CREATE MAILBOX  
      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> :REGISTERS R2-R11  
GSYSSRV CREPRC,K,12,- :CREATE PROCESS  
      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> :REGISTERS R2-R11  
GSYSSRV CREIVA,K,3,- :CREATE VIRTUAL ADDRESS  
      <R2,R3,R4,R5,R6,R7,R8>,- :REGISTERS R2-R8  
      EXC MASK :EXCEPTION MASK  
GSYSSRV DACEFC,K,1,- :DISASSOCIATE EVENT FLAG CLUSTER  
      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> :REGISTERS R2-R11  
GSYSSRV DALLOC,K,2,- :DEALLOCATE DEVICE  
      <R2,R3,R4,R5,R8> :REGISTERS R2-R5,R8  
GSYSSRV DASSGN,K,1,- :DEASSIGN I/O CHANNEL  
      <R2,R3,R4,R5,R6,R7,R8> :REGISTERS R2-R8  
GSYSSRV DCLAST,K,3,- :DECLARE AST SYSTEM SERVICE
```

```
0000 1270          <R2,R3,R4,R5>          ;REGISTERS R2-R5
0000 1271 GSYSSRV DCLXK,K,1,-             ;DECLARE EXIT HANDLER
0000 1272          <R2,R3,R4>             ;REGISTERS R2-R4
0000 1273 GSYSSRV DELLOG,ALL,3,-          ;DELETE LOGICAL NAME
0000 1274          <R2,R3,R4,R5,R6,R7,R8> ;REGISTERS R2-R8
0000 1275 GSYSSRV DELMBX,K,1,-           ;DELETE MAILBOX
0000 1276          <R2,R3,R4,R5>          ;REGISTERS R2-R5
0000 1277 GSYSSRV DELPRC,K,2,-           ;DELETE PROCESS
0000 1278          <R2,R3,R4,R5,R6,R7>    ;REGISTERS R2-R5
0000 1279 GSYSSRV DELTVA,K,3,-          ;DELETE VIRTUAL ADDRESS
0000 1280          <R2,R3,R4,R5,R6,R7>,-   ;REGISTERS R2-R7
0000 1281          EXC MASK                ;EXCEPTION MASK
0000 1282 GSYSSRV DGBLSC,K,3,-           ;DELETE GLOBAL SECTION
0000 1283          <R2,R3,R4,R5,R6,R7,R8,R9,R10> ;REGISTERS R2-R10
0000 1284 GSYSSRV DLCDNP,K,2,-           ;DEALLOCATE DIAGNOSTIC PAGE
0000 1285          <R2,R3,R4,R5,R6,R7>    ;REGISTERS R2-R7
0000 1286 GSYSSRV DLCEFC,K,1,-           ;DELETE COMMON EVENT CLUSTER
0000 1287          <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000 1288 GSYSSRV UPDSEC,K,8,-           ;UPDATE SECTION FILE
0000 1289          <R2,R3,R4,R5,R6,R7,R8>  ;R2-R8
0000 1290 GSYSSRV SNDERR,K,1,-           ;SEND MSG TO ERROR LOGGER
0000 1291          <R2,R3,R4,R5>           ;REGISTERS R2-R5
0000 1292 GSYSSRV EXIT,K,1,-             ;IMAGE EXIT
0000 1293          <R4>,0                 ;REGISTER R4, ALWAYS ALLOWED!
0000 1294 GSYSSRV EXPREG,K,4,-           ;EXPAND PROGRAM REGION
0000 1295          <R2,R3,R4,R5,R6,R7,R8>  ;REGISTERS R2-R8
0000 1296 GSYSSRV FAO,ALL,0,-            ;FORMAT ASCII OUTPUT
0000 1297          <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000 1298 GSYSSRV FAOL,ALL,0,-           ;FORMAT ASCII OUTPUT WITH VALUE LIST
0000 1299          <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000 1300 GSYSSRV FORCEX,K,3,-           ;FORCE EXIT
0000 1301          <R2,R3,R4,R5>          ;REGISTERS R2-R5
0000 1302 GSYSSRV IMGSTA,ALL,6,-         ;IMAGE STARTUP
0000 1303          <>                     ;REGISTERS NONE
0000 1304 GSYSSRV SNLIBC,E,7,-           ;SEND TO JOB CONTROLLER
0000 1305          <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000 1306 GSYSSRV GETTIM,E,1,-           ;GET TIME
0000 1307          <>                     ;NO REGISTERS
0000 1308 GCOMPSRVB UPDSECW,-            ;UPDATE SECTION AND WAIT
0000 1309          <UPDSEC_MASK ! GETJPI_SYNCH_MASK>
0000 1310 GCOMPSRVE 1
0000 1311 GSYSSRV HIBER,K,0,-              ;HIBERNATE
0000 1312          <R2,R3,R4,R5>             ;REGISTERS R2-R5
0000 1313 GSYSSRV IMGACT,E,8,-            ;IMAGE ACTIVATION
0000 1314          <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000 1315 GSYSSRV LCKPAG,K,3,-           ;LOCK PAGE IN MEMORY
0000 1316          <R2,R3,R4,R5,R6,R7,R8>  ;REGISTERS R2-R8
0000 1317 GSYSSRV LKWSET,K,3,-           ;LOCK PAGES IN WORKING SET
0000 1318          <R2,R3,R4,R5,R6,R7,R8>  ;REGISTERS R2-R8
0000 1319 GSYSSRV MGBLSC,K,7,-           ;MAP GLOBAL SECTION
0000 1320          <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000 1321 GSYSSRV PURGWS,K,1,-           ;PURGE WORKING SET
0000 1322          <R2,R3,R4,R5,R6,R7,R8>  ;R2-R8
0000 1323 GSYSSRV NUMTIM,E,2,-            ;CONVERT TIME TO NUMERIC
0000 1324          <R2,R3,R4,R5,R6,R7>    ;REGISTERS R2-R7
0000 1325 GSYSSRV SNDOPR,E,2,-          ;SEND MSG TO OPERATOR
0000 1326          <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
```



0000	1334	GSYSSRV	QIO,K,12,-	:QUEUE I/O REQUEST
0000	1335		<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>	:REGISTERS R2-R11
0000	1336	GSYSSRV	READEF,K,2,-	:READ EVENT FLAG
0000	1337		<R2,R3,R4,R5>	:REGISTERS R2-R5
0000	1338	GSYSSRV	RESUME,K,2,-	:RESUME PROCESS
0000	1339		<R2,R3,R4,R5>	:REGISTERS R2-R5
0000	1340	GSYSSRV	RUNDWN,K,1,-	:RUNDOWN
0000	1341		<R2,R3,R4,R5,R6,R7>	:REGISTERS R2-R7
0000	1342	GSYSSRV	SND\$MB,E,2,-	:SEND MSG TO SYMBIONT MANAGER
0000	1343		<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>	:REGISTERS R2-R11
0000	1344	GSYSSRV	SCHDWK,K,2,-	:SCHEDULE WAKEUP
0000	1345		<R2,R3,R4,R5,R6,R7,R8,R9>	:REGISTERS R2-R9
0000	1346	GSYSSRV	SETAST,K,1,-	:SET AST ENABLE SERVICE
0000	1347		<R2,R3,R4,R5>	:REGISTERS R2-R5
0000	1348	GSYSSRV	SETEF,K,1,-	:SET EVENT FLAG
0000	1349		<R2,R3,R4,R5>	:REGISTERS R2-R5. SEE WAITFR COMMENTS.
0000	1350	GSYSSRV	SETEXV,K,2,-	:SET EXCEPTION VECTOR
0000	1351		<R2,R3,R4,R5>	:REGISTERS R2-R5
0000	1352	GSYSSRV	SETPRN,K,1,-	:SET PROCESS NAME
0000	1353		<R2,R3,R4,R5,R6,R7,R8,R9>	:REGISTERS R2-R9
0000	1354	GSYSSRV	SETPRA,K,2,-	:SET POWER RECOVERY AST
0000	1355		<R2,R3,R4,R5>	:REGISTERS R2-R5
0000	1356	GSYSSRV	SETIMR,K,2,-	:SET TIMER
0000	1357		<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>	:REGISTERS R2-R11
0000	1358	GSYSSRV	SETPRI,K,2,-	:SET PROCESS PRIORITY
0000	1359		<R2,R3,R4,R5>	:REGISTERS R2-R5
0000	1360	GSYSSRV	SETPRT,K,5,-	:SET PAGE PROTECTION
0000	1361		<R2,R3,R4,R5,R6,R7,R8,R9>	:REGISTERS R2-R9
0000	1362	GSYSSRV	SETRWM,K,1,-	:SET RESOURCE WAIT MODE
0000	1363		<R4>	:REGISTER R4
0000	1364	GSYSSRV	SETSFM,K,1,-	:SET SYSTEM SERVICE FAILURE MODE
0000	1365		<R4>,EXC MASK	:REGISTER R4, AND EXECPTION MASK
0000	1366	GSYSSRV	SETSWM,K,1,-	:SET PROCESS SWAP MODE
0000	1367		<R4>	:REGISTER R4
0000	1368	GSYSSRV	SUSPND,K,2,-	:SUSPEND PROCESS
0000	1369		<R2,R3,R4,R5>	:REGISTERS R2-R5
0000	1370	GSYSSRV	TRNLOG,ALL,6,-	:TRANSLATE LOGICAL NAME
0000	1371		<R2,R3,R4,R5,R6,R7,R8>	:REGISTERS R2-R8
0000	1372	GSYSSRV	ULKPAG,K,3,-	:UNLOCK PAGE FROM MEMORY
0000	1373		<R2,R3,R4,R5,R6,R7,R8>	:REGISTERS R2-R8
0000	1374	GSYSSRV	ULWSET,K,3,-	:UNLOCK PAGES FROM WORKING SET
0000	1375		<R2,R3,R4,R5,R6,R7,R8>	:REGISTERS R2-R8
0000	1376	GSYSSRV	UNWIND,ALL,2,-	:UNWIND PROCEDURE CALL STACK
0000	1377		<R2,R3,R4,R5>	:REGISTERS R2-R5
0000	1378	GSYSSRV	WAITFR,K,1,-	:WAIT FOR EVENT FLAG
0000	1379		<R2,R3,R4,R5,R6>	:REGISTERS R2-R6. IF R8 IS EVER USED
0000	1380			:THE RMS SYNCHRONIZATION CODE MUST BE
0000	1381			:MODIFIED TO SAVE IT ALSO.
0000	1382	GSYSSRV	WAKE,K,2,-	:WAKE PROCESS
0000	1383		<R2,R3,R4,R5>	:REGISTERS R2-R5
0000	1384	GSYSSRV	WFLAND,K,2,-	:WAIT FOR LOGICAL AND OF EVENT FLAGS
0000	1385		<R2,R3,R4,R5,R6>	:REGISTERS R2-R6
0000	1386	GSYSSRV	WFLOR,K,2,-	:WAIT FOR LOGICAL OR OF EVENT FLAGS
0000	1387		<R2,R3,R4,R5,R6>	:REGISTERS R2-R5
0000	1388	GSYSSRV	BRDCSI,ALL,2,-	:BROADCAST TO TERMINALS
0000	1389		<R2,R3,R4,R5,R6>	:REGISTERS R2-R6
0000	1390	GSYSSRV	DCLCMH,K,3,-	:DECLARE CHANGE MODE HANDLER

0000	1391		<R4>		:SAVE R4
0000	1392	GSYSSRV	SETPFM,K,4,-		:SET PAGE FAULT MONITORING
0000	1393		<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>		:REGISTERS R2-R11
0000	1394	GSYSSRV	GETMSG,ALL,5,-		:GET MESSAGE
0000	1395		<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>		:REGISTERS R2-R11
0000	1396	GSYSSRV	DERLMB,K,1,-		:DECLARE ERROR LOG MAILBOX
0000	1397		<R2,R3,R4,R5>		:REGISTERS R2-R5
0000	1398	GSYSSRV	CANEXH,K,1,-		:CANCEL EXIT HANDLER
0000	1399		<R2,R3,R4,R5>		:REGISTERS R2-R5
0000	1400	GSYSSRV	GETCHN,K,5,-		:GET CHANNEL INFORMATION
0000	1401		<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>		:REGISTERS R2-R11
0000	1402	GSYSSRV	GETDEV,K,5,-		:GET DEVICE INFORMATION
0000	1403		<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>		:REGISTERS R2-R11
0000	1404	GSYSSRV	GETJPI,K,7,-		:GET JOB PROCESS INFORMATION
0000	1405		<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>		:REGISTERS R2-R11
0000	1406	GSYSSRV	PUTMSG,ALL,3,-		:PUT FORMATTED ERROR MESSAGE
0000	1407		<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>		:REGISTERS R2-R11
0000	1408	GSYSSRV	EXCMMSG,ALL,2,-		:OUTPUT EXCEPTION SUMMARY MESSAGE
0000	1409		<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>		:REGISTERS R2-R11
0000	1410	GSYSSRV	SNDACC,E,2,-		:SEND MSG TO ACCOUNTING MANAGER
0000	1411		<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>		:REGISTERS R2-R11
0000	1412	GSYSSRV	SETIME,K,1,-		:SET SYSTEM TIME
0000	1413		<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>		:REGISTERS R2-R11
0000	1414	GSYSSRV	SETPRV,K,4,-		:SET PRIVILEGES
0000	1415		<R2,R3,R4,R5,R6,R7,R8>		:REGISTERS R2-R8

0000 1417 :  
0000 1418 :  
0000 1419 :  
0000 1420 :  
0000 1421 :  
0000 1422 :  
0000 1423 :  
0000 1424 :  
0000 1425 :  
0000 1732 :

SPECIAL VECTORS FOR AST DELIVERY AND CLEARING

SYSSCLRAST CLEARS THE CURRENTLY ACTIVE AST STATUS

SYSSGL ASTRET CONTAINS THE VALUE OF THE RETURN ADDRESS FROM  
THE CALL INSTRUCTION USED TO DISPATCH AN AST. THIS VALUE CAN  
BE USED WHEN SEARCHING UP THE STACK FOR THE AST CALL FRAME.



```
0000 1734      .SBTTL REGION 2 OF SYS. SERV. VECTOR DEFINITIONS
0000 1735
0000 1736
0000 1737      Note: Service codes for exec mode services in this region are
0000 1738      reserved by the offset defined above between RCASCTR and ECASCTR.
0000 1739      If the ASSUME at the end of this section breaks, the offset must
0000 1740      be increased.
0000 1741
0000 1742
0000 1743      GSYSSRV ENQ,K,11,-      : ENQUEUE
0000 1744      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> : REGISTERS R2-R11
0000 1745      GSYSSRV DEQ,K,4,-      : DEQUEUE
0000 1746      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> : REGISTERS R2-R11
0000 1747      GCOMPSRVB ENQW,-      : ENQUEUE AND WAIT
0000 1748      <ENQ_MASK ! WAITFR_MASK ! CLREF_MASK ! SETEF_MASK>
0000 1762      GCOMPSRVE 3      : RESERVE 3 QUADWORDS FOR VECTOR
0000 1763      GSYSSRV SETSSF,K,1,-   : SET SYSTEM SERVICE FILTER MASK
0000 1764      <R4>                  : REGISTER R4
0000 1765      GSYSSRV SETSTK,K,3,-   : SET STACK LIMITS
0000 1766      <R2,R3,R4>            : REGISTERS R2,R3,R4
0000 1767      GSYSSRV GETSYI,K,7,-   : GET SYSTEM INFORMATION
0000 1768      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> : REGISTERS R2-R11
0000 1769      GSYSSRV IMGFIX,ALL,0,- : IMAGE ADDRESS RELOCATION FIXUP
0000 1770      <R2,R3,R4,R5>        : REGISTERS R2-R5
0000 1771      GCOMPSRVB IMGFIX_2,-  : ***** TEMP *****
0000 1772      <0>
0000 1773      GCOMPSRVE 1      : ***** TEMP *****
0000 1774      GSYSSRV GETDVI,K,8,-   : GET DEVICE AND VOLUME INFORMATION
0000 1775      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> : REGISTERS R2-R11
0000 1776      GCOMPSRVB GETDVIW,-  : GET DEVICE INFORMATION AND WAIT
0000 1777      <GETDVI_MASK ! GETJPI_SYNCH_MASK>
0000 1786      GCOMPSRVE 1      :
0000 1787      GCOMPSRVB GETJPIW,-    : GET JOB/PROCESS INFORMATION AND WAIT
0000 1788      <GETJPI_MASK ! GETJPI_SYNCH_MASK>
0000 1798      GCOMPSRVE 2      :
0000 1799      GCOMPSRVB GETSYIW,-  : GET SYSTEM INFORMATION AND WAIT
0000 1800      <GETSYI_MASK ! GETJPI_SYNCH_MASK>
0000 1809      GCOMPSRVE 1      :
0000 1810      GCOMPSRVB SNDJBCW,-  : SEND TO JOB CONTROLLER AND WAIT
0000 1811      <SNDJBC_MASK ! GETJPI_SYNCH_MASK>
0000 1820      GCOMPSRVE 1      :
0000 1821      GCOMPSRVB SYNCH,-    : SYNCHRONIZE EFN AND IOSB
0000 1822      <WAITFR_MASK ! CLREF_MASK ! SETEF_MASK>
0000 1861      GCOMPSRVE 6      : RESERVE 6 QUADWORDS FOR VECTOR
0000 1862      GSYSSRV ERAPAT,K,3,- : GENERATE A SECURITY ERASE PATTERN
0000 1863      <R4>                  : SAVE R4
0000 1864      GSYSSRV CRELNT,K,8,- : CREATE LOGICAL NAME TABLE
0000 1865      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> : REGISTERS R2-R11
0000 1866      GSYSSRV CRELNM,K,5,-  : CREATE LOGICAL NAME
0000 1867      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> : REGISTERS R2-R11
0000 1868      GSYSSRV DELLNM,K,3,-  : DELETE LOGICAL NAME
0000 1869      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> : REGISTERS R2-R11
0000 1870      GSYSSRV TRNLNM,K,5,-  : TRANSLATE LOGICAL NAME
0000 1871      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> : REGISTERS R2-R11
0000 1872      GSYSSRV GETLKI,K,7,-   : GET LOCK INFORMATION
0000 1873      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> : REGISTERS R2-R11
0000 1874      GCOMPSRVB GETLKIW,-  : GET LOCK INFORMATION AND WAIT
```

```
0000 1875 <GETLKI_MASK ! WAITFR_MASK ! CLREF_MASK ! SETEF_MASK>
0000 1887 GCOMPSRVE -2 ; RESERVE 2 QUADWORDS FOR VECTOR
0000 1888
0000 1889 GSYSSRV ASCTOID,E,3,- ; ASCII TO IDENTIFIER CONVERSION
0000 1890 <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
0000 1891 GSYSSRV FINISH_RDB,E,1,- ; FINISH RDB CONTEXT STREAM
0000 1892 <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
0000 1893 GSYSSRV IDTOASC,E,6,- ; IDENTIFIER TO ASCII CONVERSION
0000 1894 <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
0000 1895 GSYSSRV BRKTHRU,K,11,- ; BREAK THROUGH WRITES
0000 1896 <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
0000 1897 GSYSSRV GRANTID,ALL,5,- ; GRANT IDENTIFIER TO PROCESS
0000 1898 <R2,R3> ; REGISTERS R2-R3
0000 1899 GSYSSRV REVOKID,ALL,5,- ; REVOKE IDENTIFIER FROM PROCESS
0000 1900 <R2,R3> ; REGISTERS R2-R3
0000 1901 GSYSSRV CHKPRO,K,1,- ; GENERAL PROTECTION CHECK ROUTINE
0000 1902 <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
0000 1903 GCOMPSRVB BRKTHRU,- ; BREAK THOUGH WRITE AND WAIT
0000 1904 <BRKTHRU_MASK ! GETJPI_SYNCH_MASK>
0000 1913 GCOMPSRVE 2
0000 1914 GSYSSRV GETQUI,E,7,- ; GET QUEUE INFORMATION
0000 1915 <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTERS R2-R11
0000 1916 GCOMPSRVB GETQUIW,- ; GET QUEUE INFORMATION AND WAIT
0000 1917 <GETQUI_MASK ! GETJPI_SYNCH_MASK>
0000 1926 GCOMPSRVE -2
0000 1927
0000 1928 :
0000 1929 :
0000 1930 :
0000 1931 :
0000 1932 :
0000 1933 :
0000 1934 :
0000 1935 :
0000 1936 :
0000 1937 :
0000 1938 :
0000 1939 :
0000 1940 :
0000 1941 :
0000 1942 :
0000 1943 :
0000 1944 :
0000 1945 :
0000 1946 :
0000 1947 :
0000 1948 :
0000 1949 :
0000 1950 :
0000 1951 :
0000 1952 :
0000 1953 :
0000 1954 :
0000 1955 :
0000 1956 :
0000 1957 :
0000 1958 :
```

00004028

```
CJFSKASCTR = 16424

LDBSRV CJFS, ALLJDR, K, <R4>
LDBSRV CJFS, ASSJNL, K, <R4>
LDBSRV CJFS, CONUIC, K, <R4>
LDBSRV CJFS, CREJNL, K, <R4>
LDBSRV CJFS, DEALJDR, K, <R4>
LDBSRV CJFS, DEASJNL, ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
LDBSRV CJFS, DEASJNL_INT, K, <R4>
LDBSRV CJFS, DELJNL, K, <R4>
LDBSRV CJFS, DMTJMD, K, <R4>
LDBSRV CJFS, DSPJNL, K, <R4>
LDBSRV CJFS, GETJNL, K, <R4>
LDBSRV CJFS, GETRUI, K, <R4>
LDBSRV CJFS, MODFLT, K, <R4>
LDBSRV CJFS, POSJNL, K, <R4>
LDBSRV CJFS, READJNL, K, <R4>
LDBSRV CJFS, RECOVER, K, <R4>
LDBSRV CJFS, MNTJMD, K, <R4>
LDBSRV CJFS, CRENWV, K, <R4>
LDBSRV CJFS, CONJNLF, K, <R4>
LDBSRV CJFS, DCNJNLF, K, <R4>
LDBSRV CJFS, FORCEJNL, ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
LDBSRV CJFS, FORCEJNLW, ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
LDBSRV CJFS, WRITEJNL, ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
LDBSRV CJFS, WRITEJNLW, ALL, <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
LDBSRV CJFS, GETCJI, K, <R4>
LDBSRV CJFS, DMTJMDW, K, <R4>, 4, 5, DMTJMD
LDBSRV CJFS, MODFLTW, K, <R4>, 4, 5, MODFLT
LDBSRV CJFS, POSJNLW, K, <R4>, 4, 5, POSJNL
```

```
0000 1959      LDBSRV CJF$, READJNLW,   K,   <R4>, 4, 5, READJNL
0000 1960      LDBSRV CJF$, RECOVERW,   K,   <R4>, 5, 6, RECOVER
0000 1961
0000 1962 :
0000 1963 :      RUF$KCASTR = 16400
0000 1964 :
0000 1965      LDBSRV RUF$, REENTERRU,   K,   <R2,R3,R4,R5,R6>
0000 1966      LDBSRV RUF$, STARTRU,     K,   <R2,R3,R4,R5,R6>
0000 1967      LDBSRV RUF$, PHASE1,      K,   <R2,R3,R4,R5,R6>
0000 1968      LDBSRV RUF$, PHASE2,      K,   <R2,R3,R4,R5,R6>
0000 1969      LDBSRV RUF$, CANCELRU,    K,   <R2,R3,R4,R5,R6>
0000 1970      LDBSRV RUF$, MARKPOINTRU, K,   <R2,R3,R4,R5,R6>
0000 1971      LDBSRV RUF$, RESETRU,     K,   <R2,R3,R4,R5,R6>
0000 1972      LDBSRV RUF$, DCLRUI,      K,   <R2,R3,R4,R5,R6>
0000 1973      LDBSRV RUF$, CANRUH,      K,   <R2,R3,R4,R5,R6>
0000 1974      LDBSRV RUF$, RUSTATUS,    K,   <R2,R3,R4,R5,R6>
0000 1975 :
0000 1976 :      End Recovery Unit consists of a two-phase commit, so we call each
0000 1977 :      phase separately.
0000 1978 :
0000 1979      GCOMPSRVB ENDRU, <PHASE1_MASK ! PHASE2_MASK>, RUF$ ; End Recovery Unit
0000 1990      GCOMPSRVE 2
0000 1991      GSYSSRV MTACCESS,K,6,- ;Mag tape installation specific access routi
0000 1992      <R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTERS R2-R11
0000 1993
0000 1994 :
0000 1995 :      End of system service vector definitions. New system services are
0000 1996 :      to be added at this point.
0000 1997 :
0000 2003
```



```
00000053 2167 .PSECT MPSCMOD2,BYTE
00000053 2169 SSFAILMAIN: ;SSFAIL MAIN LOGIC
51 00000000'GF D0 0053 2170 MOVL G^CTL$GL PCB,R1 ;GET PCB ADDRESS
    0E A1 B5 005A 2171 TSTW PCBSW_MTXCNT(R1) ;MUTEX COUNT ZERO?
    47 12 005D 2172 BNEQ 20$ ;IF NEQ NO
    02 18 EF 005F 2173 EXTZV #PSL$V_CURMOD,#PSL$S_CURMOD,- ;EXTRACT PREVIOUS MODE FROM
    7E 04 AE 0062 2174 4(SP),=(SP) ;SAVED PSL
    6E 06 C0 0065 2175 ADDL #PCBSV_SSFEXC,(SP) ;ADD IN BASE BIT NUMBER
    24 A1 8E 0068 2176 BBC (SP)+,PCBSL_STS(R1),10$ ;IF CLEAR, FAILURE EXCEPTION DISABLED
    7E 7E DC 006D 2177 MOVPSL -(SP) ;GET CURRENT PSL
8E 6E 02 18 EF 006F 2178 EXTZV #PSL$V_CURMOD,#PSL$S_CURMOD,(SP),(SP)+ ;IF CURRENT MODE IS
    03 12 0074 2179 BNEQ 5$ ;NOT KERNEL, THEN BRANCH
    079 2180 SETIPL #0 ;FORCE IPL TO 0 FOR ERROR PATH
    0092 2191 5$: IFPRIMARY <JMP G^EXE$$$FAIL> ;IF PRIMARY, THEN CONTINUE RIGHT ALONG
    0092 2192 EXTZV #PSL$V_CURMOD,#PSL$S_CURMOD,4(SP),-(SP) ;CREATE PSL WITH PREV
    6E 6E 9C 0098 2193 ROTL #PSL$V_PRVMOD,(SP),(SP) ;MODE CORRECT AND CURRENT MODE = KERNEL
    00000000'GF 9F 009C 2194 PUSHAB G^EXE$$$FAIL ;REFLECT THE EXCEPTION
    FF5B' 31 00A2 2195 BRW MPSS$MPSCHED2 ;AND RETURN PROCESS TO PRIMARY
    02 00A5 2196 10$: REI ;RETURN FROM SERVICE WITH ERROR STATUS
    00A6 2197 20$: IFPRIMARY <BUG CHECK MTXCNTNONZ,FATAL> ;PRIMARY VERSION OF BUGCHECK
    00BD 2198 SECBUG CHECK MTXCNTNONZ,FATAL ;MUTEX COUNT NONZERO AT SERVICE EXIT
00000055 00C2 2265 KCASMAX=KCASCTR-2
    00C2 2266
    00C2 2269
```

MPCMOD  
V04-000

M 16  
- MULTIPROCESSING KERNEL SYS SRV DISPATCH 16-SEP-1984 02:08:16 VAX/VMS Macro V04-00  
REGION 2 OF SYS. SERV. VECTOR DEFINITION 5-SEP-1984 03:40:37 [SYS.SRC]CMODSSDSP.MAR;1  
00C2 2345 .END

Page 28  
(2)

MPCMOD  
Symbol table

B 1  
- MULTIPROCESSING KERNEL SYS SRV DISPATCH 16-SEP-1984 02:08:16 VAX/VMS Macro V04-00  
5-SEP-1984 03:40:37 [SYS.SPC]MODSSDSP.MAR;1

Page 29  
(2)

SSARGS	= 00000008			CMKSC_DELTVA	= 0000001A	G
SSI	= 00000024			CMKSC_DEQ	= 00000049	G
ACBSB_RMOD	= 00000008			CMKSC_DERLMB	= 00000041	G
ACBSV_KAST	= 00000007			CMKSC_DGBLSC	= 0000001B	G
ACCVIO	= 00000030	R	02	CMKSC_DLCDNP	= 0000001C	G
ACCVIO1	= 0000010D	R	02	CMKSC_DLCEFC	= 0000001D	G
ADJSTK	= 00000001			CMKSC_ENQ	= 00000048	G
ADJWSL	= 00000002			CMKSC_ERAPAT	= 0000004E	G
ALCDNP	= 00000003			CMKSC_EXIT	= 00000020	G
ALLOC	= 00000004			CMKSC_EXPREG	= 00000021	G
ASCEFC	= 00000005			CMKSC_FORCEX	= 00000022	G
ASSIGN	= 00000006			CMKSC_GETCHN	= 00000043	G
ASTEXIT	= 0000010B	R	02	CMKSC_GETDEV	= 00000044	G
BRKTHRU	= 00000054			CMKSC_GETDVI	= 0000004D	G
BUGS_MTXCNTNONZ	*****	X	03	CMKSC_GETJPI	= 00000045	G
BUGS_SSRVEXCEPT	*****	X	02	CMKSC_GETLKI	= 00000053	G
CANCEL	= 00000007			CMKSC_GETPTI	= 0000000F	G
CANEXH	= 00000042			CMKSC_GETSYI	= 0000004C	G
CANTIM	= 00000008			CMKSC_HIBER	= 00000023	G
CANWAK	= 00000009			CMKSC_LCKPAG	= 00000024	G
CATO	= 00000001			CMKSC_LKWSET	= 00000025	G
CAT7	= 00000080			CMKSC_MGBLSC	= 00000026	G
CHKPRO	= 00000055			CMKSC_MTACCESS	= 00000056	G
CJFSKASCTR	= 00004028			CMKSC_PURGWS	= 00000027	G
CLREF	= 0000000D			CMKSC_QIO	= 00000028	G
CLRPAR	= 0000000B			CMKSC_READEF	= 00000029	G
CMKSC_ADJSTK	= 00000001	G		CMKSC_RESUME	= 0000002A	G
CMKSC_ADJWSL	= 00000002	G		CMKSC_RUNDWN	= 0000002B	G
CMKSC_ALCDNP	= 00000003	G		CMKSC_SCHDWK	= 0000002C	G
CMKSC_ALLOC	= 00000004	G		CMKSC_SETAST	= 0000002D	G
CMKSC_ASCEFC	= 00000005	G		CMKSC_SETEF	= 0000002E	G
CMKSC_ASSIGN	= 00000006	G		CMKSC_SETEXV	= 0000002F	G
CMKSC_BRKTHRU	= 00000054	G		CMKSC_SETIME	= 00000046	G
CMKSC_CANCEL	= 00000007	G		CMKSC_SETIMR	= 00000032	G
CMKSC_CANEXH	= 00000042	G		CMKSC_SETPFM	= 00000040	G
CMKSC_CANTIM	= 00000008	G		CMKSC_SETPRA	= 00000031	G
CMKSC_CANWAK	= 00000009	G		CMKSC_SETPRI	= 00000033	G
CMKSC_CHKPRO	= 00000055	G		CMKSC_SETPRN	= 00000030	G
CMKSC_CLREF	= 0000000D	G		CMKSC_SETPRT	= 00000034	G
CMKSC_CLRPAR	= 0000000B	G		CMKSC_SETPRV	= 00000047	G
CMKSC_CMKRN	= 0000000C	G		CMKSC_SETRWM	= 00000035	G
CMKSC_CNTREG	= 0000000E	G		CMKSC_SETSFM	= 00000036	G
CMKSC_CRELNM	= 00000050	G		CMKSC_SETSSF	= 0000004A	G
CMKSC_CRELNT	= 0000004F	G		CMKSC_SETSTK	= 0000004B	G
CMKSC_CREMBX	= 00000010	G		CMKSC_SETSWH	= 00000037	G
CMKSC_CREPRC	= 00000011	G		CMKSC_SNDERR	= 0000001F	G
CMKSC_CRETVA	= 00000012	G		CMKSC_SUSPND	= 00000038	G
CMKSC_CRMPS	= 0000000A	G		CMKSC_TRNLNM	= 00000052	G
CMKSC_DACEFC	= 00000013	G		CMKSC_ULKPAG	= 00000039	G
CMKSC_DALLOC	= 00000014	G		CMKSC_ULWSET	= 0000003A	G
CMKSC_DASSGN	= 00000015	G		CMKSC_UPDSEC	= 0000001E	G
CMKSC_DCLAST	= 00000016	G		CMKSC_WAITFR	= 0000003B	G
CMKSC_DCLCMH	= 0000003F	G		CMKSC_WAKE	= 0000003C	G
CMKSC_DCLEXH	= 00000017	G		CMKSC_WFLAND	= 0000003D	G
CMKSC_DELLNM	= 00000051	G		CMKSC_WFLOR	= 0000003E	G
CMKSC_DELMBX	= 00000018	G		CMKRNE	= 0000000C	
CMKSC_DELPRC	= 00000019	G		CNTREG	= 0000000E	



MPCMOD  
Symbol table

- MULTIPROCESSING KERNEL SYS SRV DISPATCH 16-SEP-1984 02:08:16 VAX/VMS Macro V04-00  
5-SEP-1984 03:40:37 [SYS.SRC]CMODSSDSP.MAR;1

Page 30  
(2)

CRELNM	=	00000050		
CRELNT	=	0000004F		
CREMBX	=	00000010		
CREPRC	=	00000011		
CRETVA	=	00000012		
CRMPSC	=	0000000A		
CTL\$GB_SSFILTER	=	*****	X	02
CTL\$GL_PCB	=	*****	X	02
DACEFC	=	00000013		
DALLOC	=	00000014		
DASSGN	=	00000015		
DCLAST	=	00000016		
DCLCMH	=	0000003F		
DCLEXH	=	00000017		
DEF_MASK	=	00000081		
DELNM	=	00000051		
DELMBX	=	00000018		
DELPRC	=	00000019		
DELTVA	=	0000001A		
DEQ	=	00000049		
DERLMB	=	00000041		
DGBLSC	=	0000001B		
DLCDNP	=	0000001C		
DLCEFC	=	0000001D		
ENQ	=	00000048		
ENQ\$_ACHODE	=	00000028		
ENQ\$_ASTADR	=	0000001C		
ENQ\$_ASTPRM	=	00000020		
ENQ\$_BLKAST	=	00000024		
ENQ\$_EFN	=	00000004		
ENQ\$_FLAGS	=	00000010		
ENQ\$_LKMODE	=	00000008		
ENQ\$_LKSB	=	0000000C		
ENQ\$_NARGS	=	0000000B		
ENQ\$_PARID	=	00000018		
ENQ\$_PROT	=	0000002C		
ENQ\$_RESNAM	=	00000014		
ERAPAT	=	0000004E		
EXC_MASK	=	00000080		
EXESMODKRN	=	*****	X	02
EXESGL_RPB	=	*****	X	02
EXESREFLECT	=	*****	X	02
EXESSFAIL	=	*****	X	03
EXIT	=	00000020		
EXPREG	=	00000021		
FORCEX	=	00000022		
GETCHN	=	00000043		
GETDEV	=	00000044		
GETDVI	=	0000004D		
GETDVIS_ASTADR	=	00000018		
GETDVIS_ASTPRM	=	0000001C		
GETDVIS_CHAN	=	00000008		
GETDVIS_DEVNAM	=	0000000C		
GETDVIS_EFN	=	00000004		
GETDVIS_IOSB	=	00000014		
GETDVIS_ITMLST	=	00000010		
GETDVIS_NARGS	=	0000000B		

GETDVIS_NULLARG	=	00000020		
GETJPI	=	00000045		
GETJPI\$ASTADR	=	00000018		
GETJPI\$ASTPRM	=	0000001C		
GETJPI\$EFN	=	00000004		
GETJPI\$IOSB	=	00000014		
GETJPI\$ITMLST	=	00000010		
GETJPI\$NARGS	=	00000007		
GETJPI\$PIDADR	=	00000008		
GETJPI\$PRCNAM	=	0000000C		
GETLKI	=	00000053		
GETLKI\$ASTADR	=	00000014		
GETLKI\$ASTPRM	=	00000018		
GETLKI\$EFN	=	00000004		
GETLKI\$IOSB	=	00000010		
GETLKI\$ITMLST	=	0000000C		
GETLKI\$LKIDADR	=	00000008		
GETLKI\$NARGS	=	00000007		
GETLKI\$RESERVED	=	0000001C		
GETPTI	=	0000000F		
GETSYI	=	0000004C		
GETSYI\$ASTADR	=	00000018		
GETSYI\$ASTPRM	=	0000001C		
GETSYI\$CSIDADR	=	00000008		
GETSYI\$EFN	=	00000004		
GETSYI\$IOSB	=	00000014		
GETSYI\$ITMLST	=	00000010		
GETSYI\$NARGS	=	00000007		
GETSYI\$NODENAME	=	0000000C		
HIBER	=	00000023		
INHEXCP	=	00000000	R	02
IPL\$SYNCH	=	00000008		
KCASCTR	=	00000057		
KCASE	=	0000011C	R	02
KCASMAY	=	00000055		
KERDSP	=	000000B7	R	02
KINSARG	=	0000003E	R	02
KINSARG1	=	00000110	R	02
LCKSV_INTERLOCK	=	00000000		
LCKPAG	=	00000024		
LKWSET	=	00000025		
MGBLSC	=	00000026		
MP\$ASTEXIT	=	00000000	R	03
MP\$CMODKRN	=	00000088	RG	02
MP\$CMODKRN1X	=	00000060	RG	02
MP\$EXCPTN	=	0000004F	RG	02
MP\$SGL_CURPCB	=	*****	X	03
MP\$SGL_INTERLOCK	=	*****	X	03
MP\$SMP\$SCHED2	=	*****	X	02
MP\$SSECBUGCHK	=	*****	X	02
MP\$SWAITFR	=	*****	X	02
MP\$SWAITFR1	=	00000113	R	02
MP\$SWFLAND	=	*****	X	02
MP\$SWFLAND1	=	00000116	R	02
MP\$SWFLOR	=	*****	X	02
MP\$SWFLOR1	=	00000119	R	02
MPSWITCH	=	00000001		

MPCMOD  
Symbol table

D 1

- MULTIPROCESSING KERNEL SYS SRV DISPATCH 16-SEP-1984 02:08:16 VAX/VMS Macro V04-00  
5-SEP-1984 03:40:37 [SYS.SRC]CMODSSDSP.MAR;1

Page 31  
(2)

MTACCESS = 00000056  
PCBSB\_ASTACT = 0000000C  
PCBSL\_ASTQFL = 00000010  
PCBSL\_PHD = 0000006C  
PCBSL\_STS = 00000024  
PCBSV\_SSFEXC = 00000006  
PCBSW\_MTXCNT = 0000000E  
PHDSB\_ASTLVL = 000000CF  
PRS\_ASTLVL = 00000013  
PRS\_IPL = 00000012  
PRS\_SCBB = 00000011  
PSL\$M\_CURMOD = 03000000  
PSL\$S\_CURMOD = 00000002  
PSL\$V\_CURMOD = 00000018  
PSL\$V\_PVRMOD = 00000016  
PURGWS = 00000027  
QIO = 00000028  
QIOS\_ASTADR = 00000014  
QIOS\_ASTPRM = 00000018  
QIOS\_CHAN = 00000008  
QIOS\_EFN = 00000004  
QIOS\_FUNC = 0000000C  
QIOS\_IOSB = 00000010  
QIOS\_NARGS = 0000000C  
QIOS\_P1 = 0000001C  
QIOS\_P2 = 00000020  
QIOS\_P3 = 00000024  
QIOS\_P4 = 00000028  
QIOS\_P5 = 0000002C  
QIOS\_P6 = 00000030  
READEF = 00000029  
RESUME = 0000002A  
RPBSL\_SCBB = 000000B0  
RUF\$K\$ASCTR = 00004010  
RUNDWN = 0000002B  
SCHDWK = 0000002C  
SETAST = 0000002D  
SETEF = 0000002E  
SETEXV = 0000002F  
SETIME = 00000046  
SETIMR = 00000032  
SETPFM = 00000040  
SETPRA = 00000031  
SETPRI = 00000033  
SETPRN = 00000030  
SETPRT = 00000034  
SETPRV = 00000047  
SETRWM = 00000035  
SETSM = 00000036  
SETSSF = 0000004A  
SETSTK = 0000004B  
SETSUM = 00000037  
SNDERR = 0000001F  
SNDJBC\$\_ASTADR = 00000018  
SNDJBC\$\_ASTPRM = 0000001C  
SNDJBC\$\_EFN = 00000004  
SNDJBC\$\_FUNC = 0000000B

SNDJBC\$\_IOSB = 00000014  
SNDJBC\$\_ITMLST = 00000010  
SNDJBC\$\_NARGS = 00000007  
SNDJBC\$\_NULLARG = 0000000C  
SRVEXIT = 0000004B R 02  
SRVREI = 0000004E R 02  
SS\$\_ACCVIO = 0000000C  
SS\$\_INHCHMK = 000004CC  
SS\$\_INSFARG = 00000114  
SSFAIL = 00000056 R 02  
SSFAILMAIN = 00000053 R 03  
SUSPND = 00000038  
SYNCH\$\_EFN = 00000004  
SYNCH\$\_IOSB = 00000008  
SYNCH\$\_NARGS = 00000002  
SYSSGB\_KMASK = \*\*\*\*\* X 02  
SYSSGB\_KRNLNARG = \*\*\*\*\* X 02  
TRNLNM = 00000052  
ULKPAG = 00000039  
ULWSET = 0000003A  
UPDSEC = 0000001E  
UPDSEC\$\_ACMODE = 0000000C  
UPDSEC\$\_ASTADR = 0000001C  
UPDSEC\$\_ASTPRM = 00000020  
UPDSEC\$\_EFN = 00000014  
UPDSEC\$\_INADR = 00000004  
UPDSEC\$\_IOSB = 00000018  
UPDSEC\$\_NARGS = 00000008  
UPDSEC\$\_RETADR = 00000008  
UPDSEC\$\_UPDFLG = 00000010  
WAITFR = 0000003B  
WAKE = 0000003C  
WFLAND = 0000003D  
WFLOR = 0000003E

+-----+  
! Psect synopsis !  
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 ( 0.)	01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
MPSCMOD1	00000120 ( 288.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC QUAD
MPSCMOD2	000000C2 ( 194.)	03 ( 3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
\$\$\$000	00000000 ( 0.)	04 ( 4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC QUAD

+-----+  
! Performance indicators !  
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.07	00:00:00.82
Command processing	157	00:00:01.25	00:00:07.91
Pass 1	668	00:00:23.94	00:00:58.52
Symbol table sort	0	00:00:02.07	00:00:03.17
Pass 2	225	00:00:06.60	00:00:20.10
Symbol table output	38	00:00:00.29	00:00:01.29
Psect synopsis output	2	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	1121	00:00:34.25	00:01:31.85

The working set limit was 2250 pages.  
210745 bytes (412 pages) of virtual memory were used to buffer the intermediate code.  
There were 70 pages of symbol table space allocated to hold 1356 non-local and 11 local symbols.  
2351 source lines were read in Pass 1, producing 23 object records in Pass 2.  
51 pages of virtual memory were used to define 47 macros.

+-----+  
! Macro library statistics !  
+-----+

Macro library name	Macros defined
_\$255\$DUA28:[MP.OBJ]MP.MLB;1	8
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	9
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	19
TOTALS (all libraries)	36

1362 GETS were required to define 36 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:MPCMOD/OBJ=OBJ\$:MPCMOD MSRC\$:MPPREFIX/UPDATE=(ENH\$:MPPREFIX)+MSRC\$:MPSWT/UPDATE=(ENH\$:MPSWT)+MASD\$: [SYS.SRC]CMODSSDSP



0247 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

VMOUNT  
LIS

MPCLRPFM  
LIS

MPAST  
LIS

MP

MP  
MAP

MP  
MDL

TRNLOG  
LIS

MPMOD  
LIS

MPMACROS  
MAR



0248 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

